

# Bases de Datos

---

Oscar Marban → 4302 → [omarban@fi.upm.es](mailto:omarban@fi.upm.es)

Apuntes de Pau Arlandis Martinez

## Contenido

1.- Introducción .....	2
1.1.- ¿Qué es una base de datos? .....	2
1.2.- Introducción al modelo relacional .....	2
1.2.1.- Relación.....	3
1.2.2.- Restricciones .....	3
1.3.- Álgebra relacional .....	5
1.3.1.- Definición .....	5
2.- Diseño conceptual.....	5
2.1.- Modelo E/R .....	6
Entidad .....	6
Atributos.....	6
Relaciones .....	7
Ejercicio de entidad/relación. ....	10
Ejercicio 2 .....	11
Ejercicio para casa .....	11
2.2.- Extensión de ER.....	11
Dependencias.....	11
Jerarquías .....	12
2.3 Históricos.....	13
3.- Transformación a SGBD .....	14
3.1.- Paso a tablas .....	14
1. Toda entidad genera una tabla. ....	14
2. Relaciones N:M.....	14
3. Relaciones 1:N .....	15
4. Relaciones 1:1 .....	15
5. Atributos multivaluados .....	16
6. Relaciones grado >2 .....	16
7. Históricos.....	16

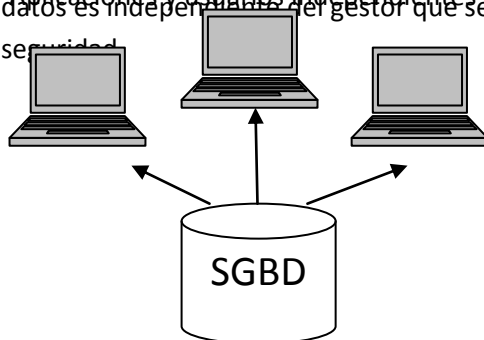
8. Jerarquías .....	16
9. Relaciones de dependencia.....	17
Apendice A: ejercicio2.dia .....	19
Apéndice B: ejerciciolunes.dia .....	20

## 1.- Introducción

### 1.1.- ¿Qué es una base de datos?

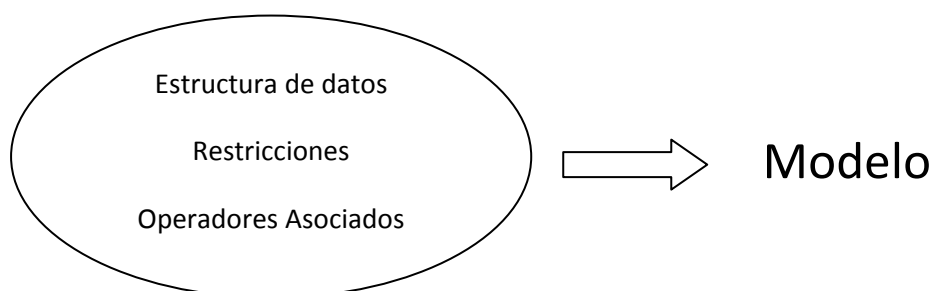
Colección o depósito de datos integrados, almacenados en soporte secundario (no volátil) y con **redundancia controlada**. Los datos, que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de ellos, y su definición (estructura de la base de datos) única y almacenada junto con los datos, se ha de apoyar en un modelo de datos, el cual ha de permitir captar las interrelaciones y de restricciones existentes en el mundo real. Los procedimientos de actualización y recuperación, comunes y bien determinados, facilitarán la seguridad del conjunto de los datos. [Miguel y Piattini, 1999]

En una base de datos, tal y como la entendemos nosotros, la redundancia debe estar controlada y minimizada. Los datos deben ser independientes de la aplicación. La estructura de datos es independiente del gestor que se utilice. También, una BD, debe poseer un sistema de seguridad.



E/R → Relacional → MySQL / Postgrés, SQL Server, Oracle, DB2.

### 1.2.- Introducción al modelo relacional



**Estructura:** Relación

**Restricciones:** { Básicas (del modelo)  
Dependencias

**Operadores:** Álgebra relacional

### 1.2.1.- Relación

$R(A,B,C)$  A, B, C se denominan atributos.

R:

A	B	C
a1	b2	c1
a3	b1	c2
a2	b2	c1
a1	b3	c3

Grado: 3  $\rightarrow$  Número de columnas.

Cardinalidad: 4  $\rightarrow$  Número de filas.

$\text{Dom}(A) = \{a1, a2, a3...\}$

$\text{Dom}(B) = \{b1, b2, b3...\}$

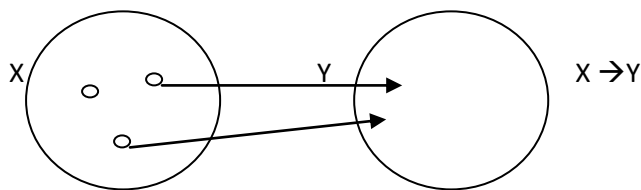
$\text{Dom}(C) = \{c1, c2, c3...\}$

$R \subset \text{Dom}(A) \times \text{Dom}(B) \times \text{Dom}(C)$

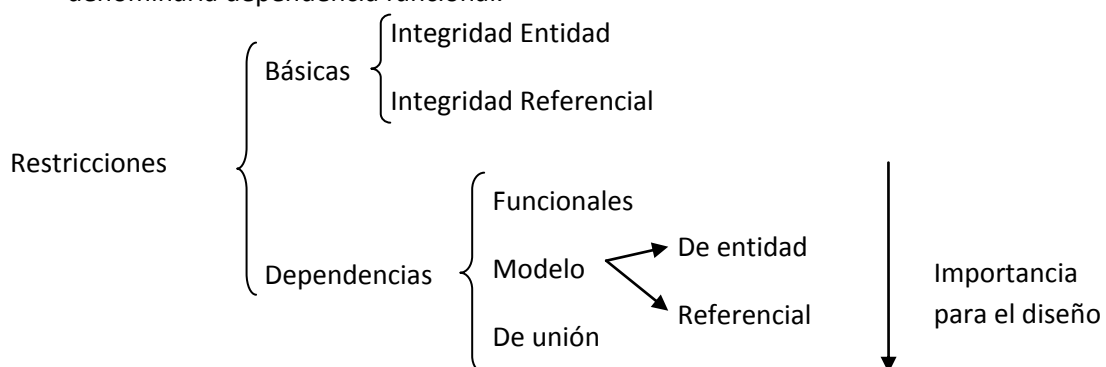
El elemento básico del diseño del modelo relacional es el atributo monovaluado.

### 1.2.2.- Restricciones

#### 1.2.2.1.- Dependencia



Cada elemento del dominio de X se corresponde a un elemento de Y. Cada elemento de X tiene un solo elemento de Y. Puede haber elementos no relacionados. Esto es lo que se denominaría dependencia funcional.



## Clave

Conjunto no red. De atributos que determinan el valor de los demás

DNI	Nombre	Dirección	Teléfono	e-mail	Puesto	Antigüedad	Salario

Descriptor es cualquier subconjunto de las claves.

Dentro de la clave, cada atributo puede poseer dependencias funcionales entre sí. Por ejemplo, el DNI podría determinar el nombre, el teléfono, etc... Las dependencias funcionales son las más importantes para determinar el diseño.

La clave debe ser no redundante. Las dependencias se definen sin tener en cuenta el contenido de la clave.

La integridad entidad y la integridad referencial dependen de la clave. Si tengo  $R(A, B, C, D)$ , una clave es un subconjunto de  $R$ . Es un conjunto de atributos que determinan una relación. En una relación existe como mínimo una clave. Las claves pueden seguir un orden jerárquico.

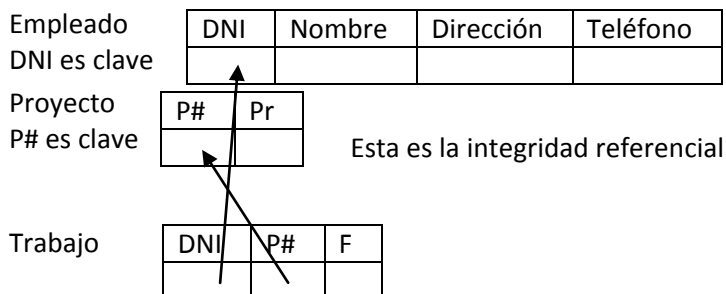
## Atributos primarios o principales

Son aquellos que forman parte de una clave.

## Atributos secundarios o no principales

No forman parte de la clave.

Ejemplos:



DNI y P# son clave

La integridad entidad se refiere a que una relación debe poseer sus claves para ser correcta. Las claves son necesarias para las relaciones.

La integridad referencial es la que hace necesaria a una tupla de una relación que es clave, haga referencia a otras relaciones donde también es clave. Es decir, en la relación trabajo las claves son DNI y P#, que son clave también de Empleado y Proyecto (respectivamente). Para que se cumpla la integridad referencial DNI y P# que aparecen en la relación trabajo aparecen también en Empleado y Proyecto.

## 1.3.- Álgebra relacional

### 1.3.1.- Definición

Operandos: Relacionales.

Operadores:

- Básicos
  - Unión
  - Diferencia
  - Producto Cartesiano
  - Proyección
  - Selección
- Derivados
  - Intersección
  - División
  - Asociación
  - Unión Natural o *Join*

Está basada en el álgebra de conjuntos (fue definida por Codd).

## 2.- Diseño conceptual.

Para solucionar cualquier problema en bases de datos se hace necesario crear un modelo que sea independiente del gestor que se utilice para esa base de datos.

La arquitectura ANSI/SPARC es una arquitectura basada en tres niveles: un modelo físico, donde se almacena la información; un modelo conceptual y, por último, un modelo externo, el nivel de los usuarios. Un sistema gestor es capaz de interactuar entre estos tres niveles. Tanto el modelo relacional como el E/R son modelos conceptuales, independientes del gestor y únicos para una base de datos. Suponen la vista del modelo de datos.

El modelo físico va a ser único e independiente del gestor.

El modelo externo es la forma en la que ven los usuarios la base de datos. Existen múltiples visiones de los datos, dependiendo de cuantos usuarios puedan ver los datos.

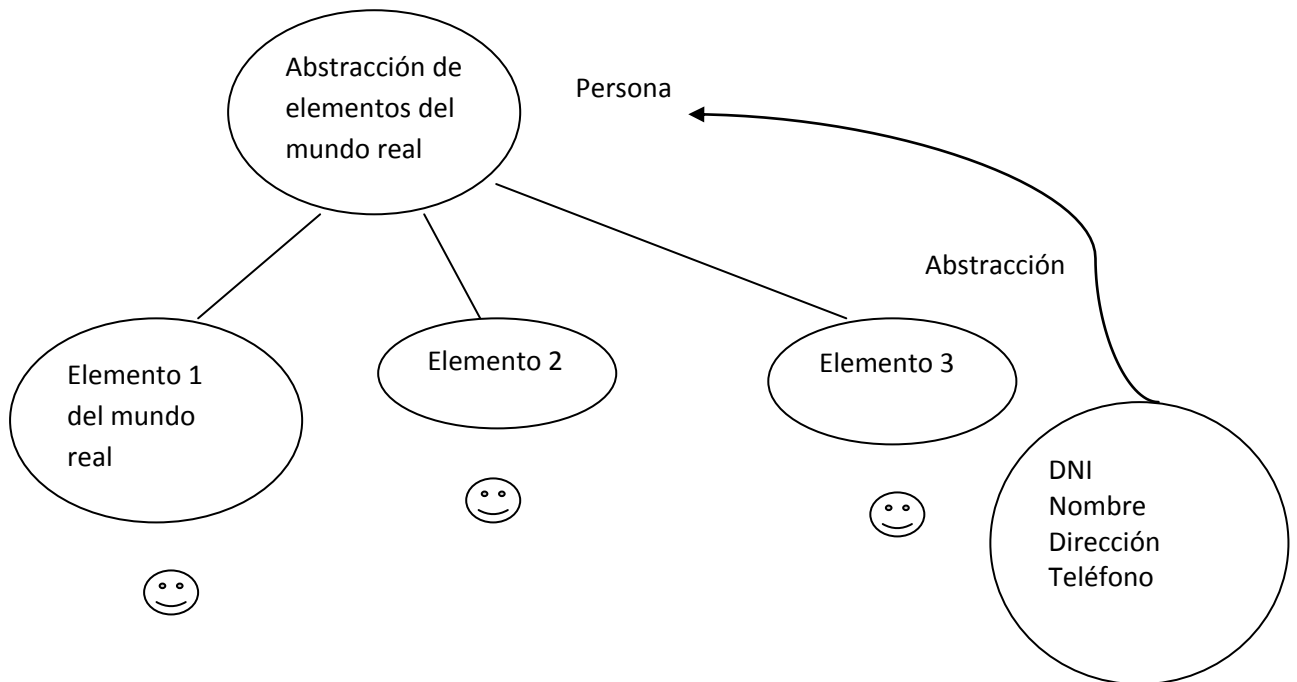
El gestor de la base de datos realiza funciones de mapeo entre cada nivel, este mapeo se almacenan en metadatos y dan al gestor toda la información necesaria de los datos.

Uno de los mejores gestores (y el que vamos a utilizar en el curso) es el lenguaje SQL. Es capaz de reconocer cualquier modelo relacional. El lenguaje SQL se divide en 3 niveles: DDL, DML y DCL. Los dos primeros tratan el mapeo entre el modelo físico y el conceptual; el último es el lenguaje de programación en sí mismo y el que realiza el mapeo entre el diseño conceptual y el externo.

## 2.1.- Modelo E/R

El modelo E/R es un modelo totalmente abstracto, completamente independiente del gestor. En él simplemente nos preguntaremos qué información vamos a almacenar. En ningún caso, cómo lo vamos a almacenar, que es competencia exclusiva del SGBD.

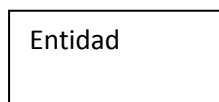
Este modelo fue propuesto en los 70 por Peter Chen, y en este curso vamos a utilizar la notación de Chen. Así que los modelos que diseñemos utilizaremos la notación de este autor. El modelo E/R pretende que de un conjunto de elementos seamos capaces de crear una única representación que almacene todas las características de elementos del mundo real:



Siempre se debe pensar qué hay que almacenar, más tarde cómo. Lo primero que debemos pensar es en las ocurrencias de los elementos del modelo.

### Entidad

Una entidad es un mecanismo de abstracción que permite definir aquellos elementos de los cuales interesa almacenar información. Su representación en un E/R es la siguiente:

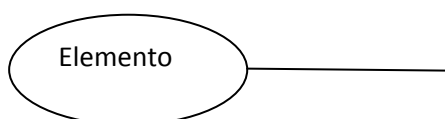


Dentro se da el nombre que se le da a ese conjunto. En nuestro problema de ejemplo la entidad es Persona.

De las entidades se almacenará una serie de elementos denominados Atributos.

### Atributos

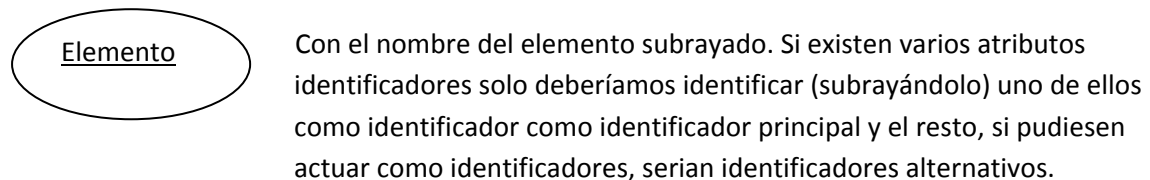
Aquella información común que quiero almacenar de los elementos de una entidad. En el modelo de E/R se representan mediante un óvalo y se unen a la entidad por una línea.



Siempre debemos tratar de evitar la duplicidad de información y el error en los datos. Se debe pensar que un atributo no almacena más información que ella misma. Si un atributo almacena información de otros atributos, es entidad.

### Atributo identificador

Como norma general las entidades tendrán al menos 3 atributos. De ellos hay un subconjunto que nos permite distinguir distintas propiedades de una entidad. Este atributo es un **atributo identificador**. Es aquel atributo que me permite diferenciar cada atributo de la entidad. Cada ocurrencia. En nuestra representación se representa de la siguiente forma:



### Identificador alternativo

No se marcan de ninguna forma, deben ser únicos y sin valor vacío.

Toda entidad tiene un identificador principal. En caso de que en el mundo real no posean tal identificador, debemos crear uno ficticio.

### Identificador compuesto

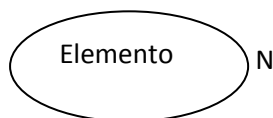
En ocasiones una entidad puede poseer diversos identificadores que actúan como identificadores primarios. Se convierten en identificadores compuestos que pueden poseer ocurrencias a nivel individual (pueden existir ocurrencias en uno o en otro) pero nunca en los dos al tiempo.

El resto de atributos de una entidad se denominarán **atributos descriptivos**.

### Atributo multivaluado

Un atributo multivaluado es uno que puede tener varios valores. Es raro de encontrar en un modelo y son cosas como el teléfono, el email, etc...

En la representación de Chen es un óvalo con el nombre del atributo dentro y una N en su superíndice, donde se representa el número de atributos.



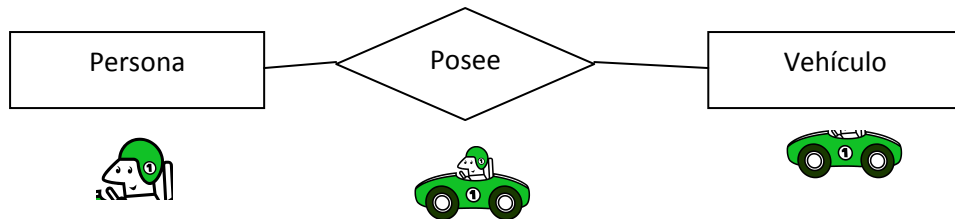
Se insiste en que los atributos multivaluados son muy raros de encontrar.

### Relaciones

Las relaciones definen uniones entre entidades. Por ejemplo, los coches que tiene un propietario, las asignaturas de un alumno, etc... En Chen se representa como un rombo que une dos entidades.



Una relación puede tener dos o más entidades relacionadas. Al número de relaciones entre entidades se le llama **grado**. El grado 2, dos entidades relacionadas, es lo más común.



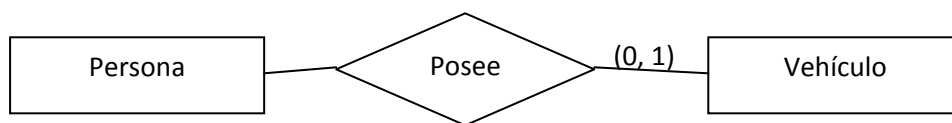
La **cardinalidad**, otra característica de las relaciones, determina el número de ocurrencias de una entidad con las que se relaciona otra entidad diferente. Una persona posee varios coches. Esta cardinalidad depende de la **correspondencia** de la relación. Existen varios tipos de ocurrencia:

**1:1** Una entidad se relaciona con una sola ocurrencia de otra. Un coche, una persona. Una persona, un coche.

**1:N** Una entidad se relaciona con varias ocurrencias de otra. Una persona, varios coches. Un coche, una persona.

**N:M** Una entidad se relaciona con varias ocurrencias de otra, en ambos sentidos. Una persona, varios coches. Un coche, varias personas.

En la representación de Chen, se coloca una tupla entre paréntesis con el mín de elementos que se pueden corresponder y el máximo. Van cruzadas, colocándose en la línea de la relación del extremo final. Si A se relaciona con B de alguna manera, la tupla se coloca sobre la línea que une la relación con B

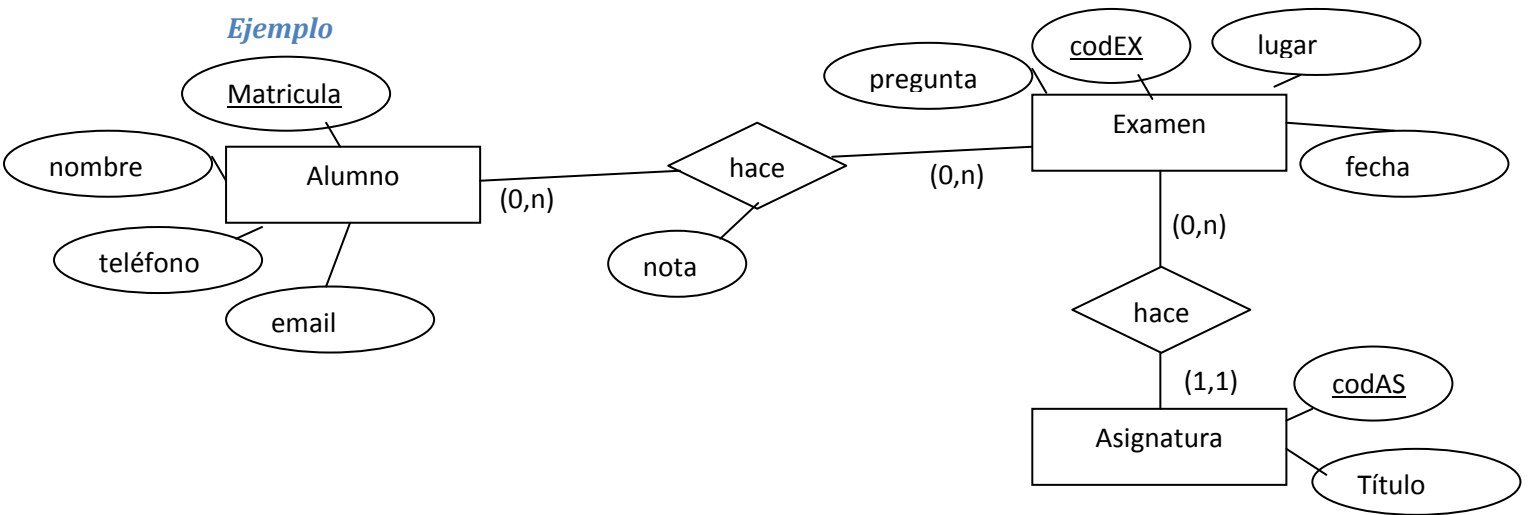


En Chen solo se permiten 0 ó 1 en el mínimo de la tupla y 1 ó n en el máximo. Así no existen más que las correspondencias  $(0,1)$ ,  $(1,1)$ ,  $(0,n)$ ,  $(1,n)$ .

Las relaciones también puede tener información adicional, atributos.

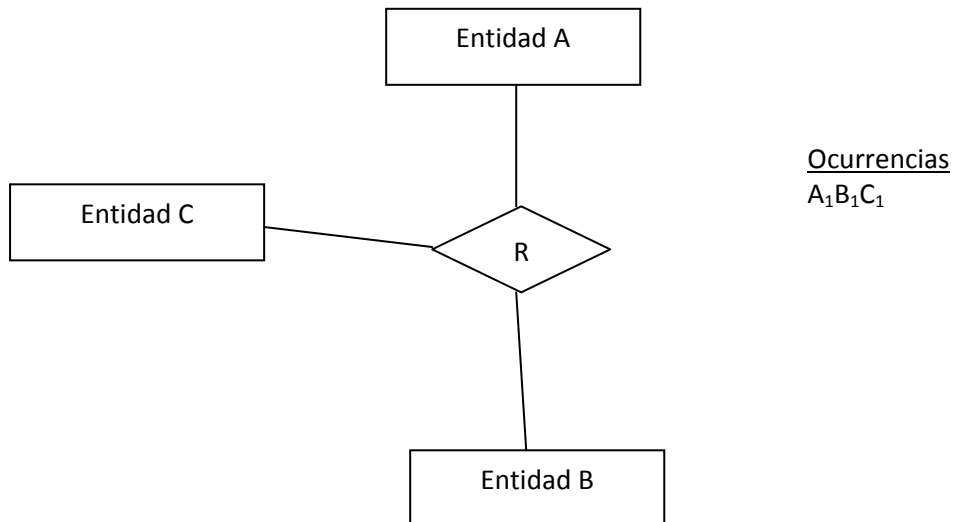


### Ejemplo



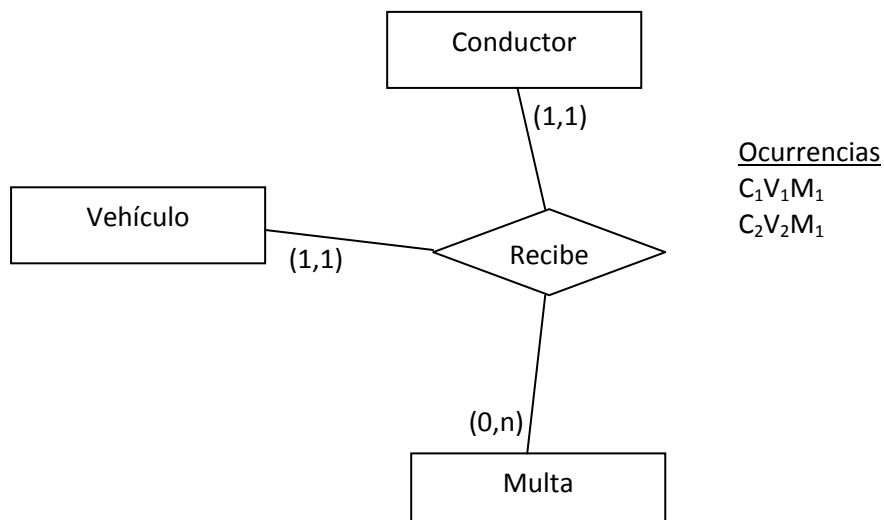
### Relaciones de grado mayor a 2

Las relaciones de grado mayor que 2 son raras, sobre todo a partir de grado 4. Hay que estar seguro de lo que se hace con ellas.

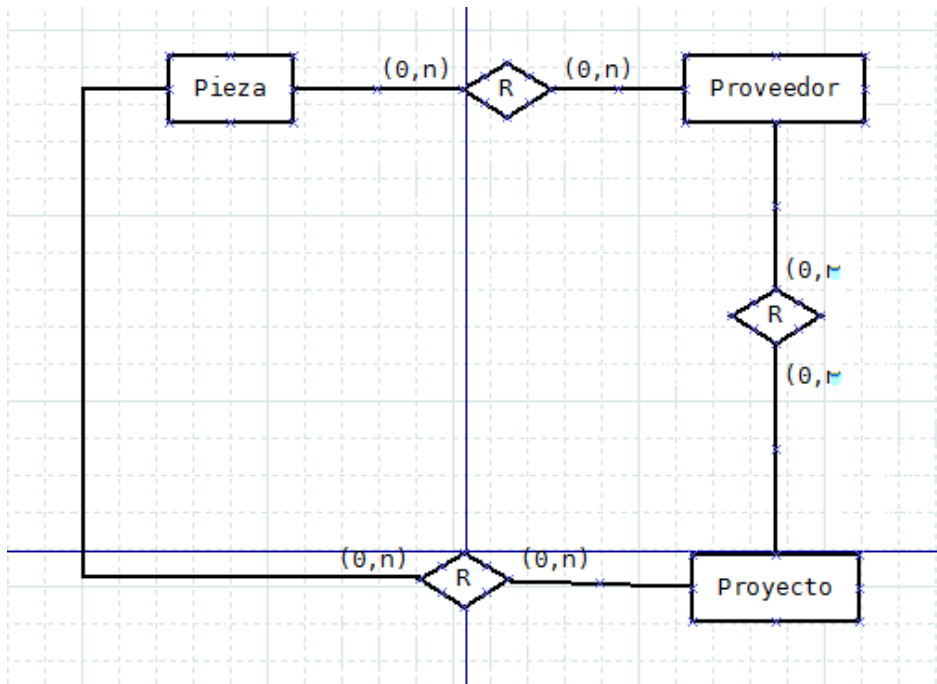


Cuando se relacionan tres elementos (o más) las ocurrencias de esa relación deben tener todas tres (o más) elementos.

Por ejemplo:



La descomposición de las relaciones ternarias dependerá de las cardinalidades. Por ejemplo, si intentáramos descomponer el ejemplo anterior en relaciones binarias



Descubriremos que estamos cayendo en un error, ya que las ocurrencias necesitan poseer los tres elementos para saber relacionarlas bien, nos faltaría información si no.

De forma genérica si se poseen solo cardinalidades  $(0,n)$  será imposible (generalmente) descomponerla y si en una relación ternaria existe una cardinalidad  $(1,1)$  entonces, por lo general, podrá descomponerse (pero no siempre).

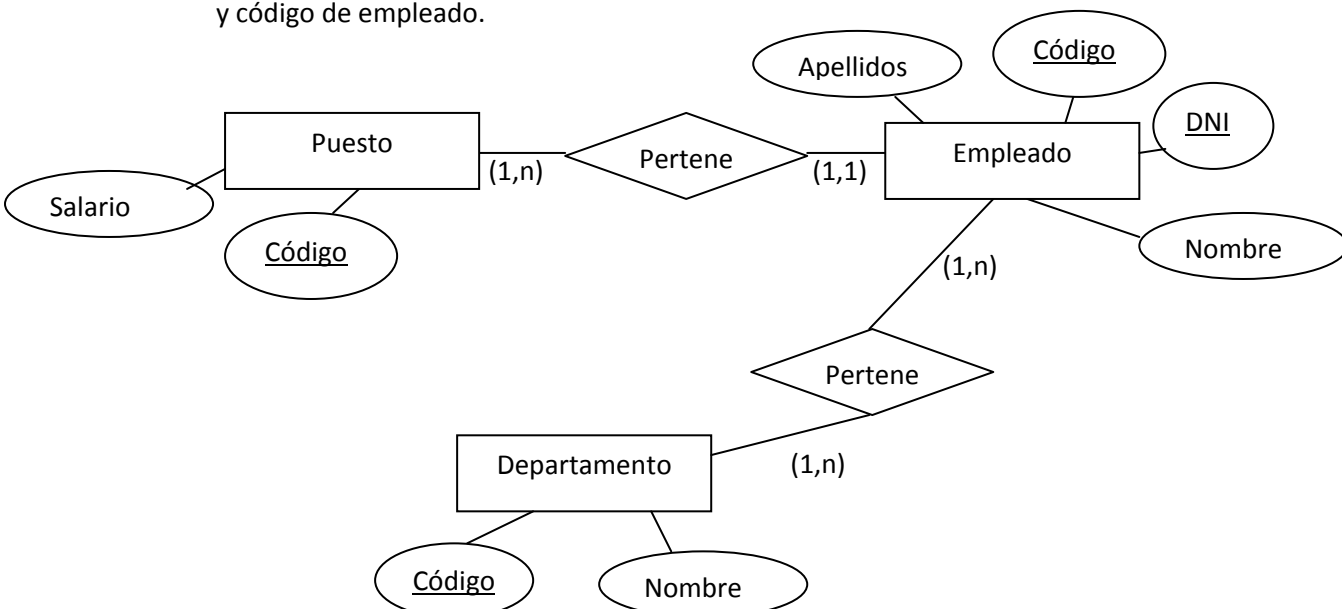
### Ejercicio de entidad/relación.

Diseñar la base de datos de una empresa

Puestos -> Código y Salario asociado

empresa -> departamentos con código, nombre y empleados

empleado -> En varios departamentos y se asocia a un departamento. Nombre, apellidos, DNI y código de empleado.



## Ejercicio 2

(Mirar [Apéndice A: ejercicio2.dia](#))

Si nos encontramos un camino cerrado puede ser un indicador de duplicación de información.

## Ejercicio para casa

(Mirar [Apéndice B: ejercicialunes.dia](#))

## 2.2.- Extensión de ER

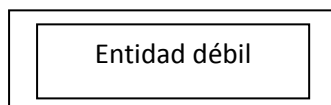
Se utiliza para extender cualquier BDER, existen dos tipos:

- Jerarquias
- Dependencias

### Dependencias

#### Entidad Débil

Las entidades débiles poseen las mismas características que una entidad excepto porque no tienen identificadores únicos y/o no tienen ocurrencias por sí mismas. Es decir, dependen de otra entidad superior, de una entidad regular. En la simbología de Chon se presentan así:



Las entidades débiles son entidades raras, hay que tener cuidado al crearlas pues al borrar la entidad fuerte de la que dependen desaparece.

#### Dependencia de existencia

Las entidades débiles no tienen ocurrencias por sí mismas, dependen de una entidad regular para existir, pero si que tienen identificadores únicos.



#### Dependencia en identificación

En este caso, no solo una entidad débil depende de otro en existencia sino que además no tienen identificadores únicos, su identificador depende del identificador de la entidad regular. Toda dependencia en identificación es a su vez de existencia.



## Jerarquías

Cuando tenemos varias entidades que comparten ciertos atributos podemos utilizar esta extensión para simplificar tomando una identidad genérica que agrupe esos atributos comunes y que se relacione con las entidades anteriores, llamadas específicas. (Ver jerarquía.dia)

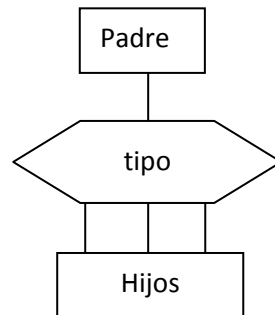
Toda ocurrencia que aparezca en vehículo aparecerá en una de las entidades específicas, y viceversa. Las identidades específicas no tienen identificador, este está en la entidad genérica.

Existen dos tipos de jerarquías:

- Subconjunto
- Generalización/Especificación

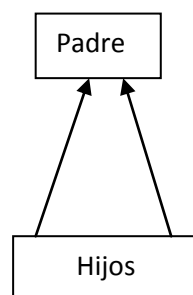
### Generalización/Especificación

Se representa de la siguiente forma:



En este caso, toda ocurrencia de vehículo va a ser de un solo tipo de sus entidades específicas.

### Subconjunto

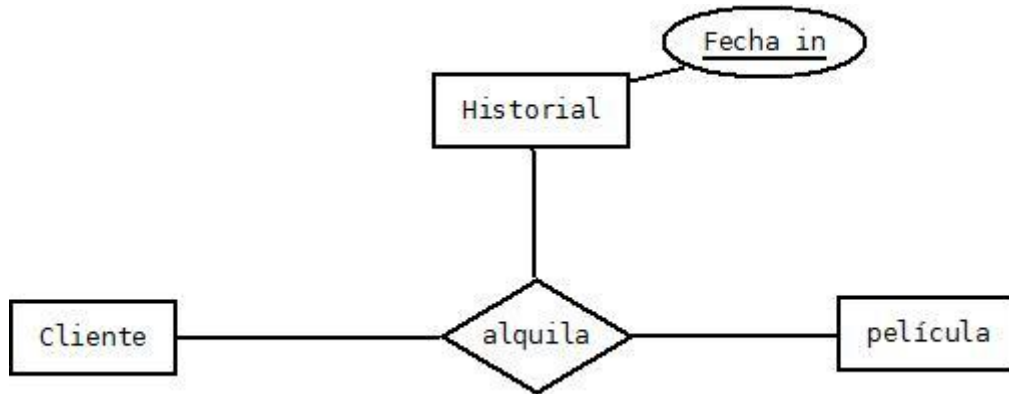


En este caso, cada ocurrencia del padre podría tener una, varias o todas de sus hijas.

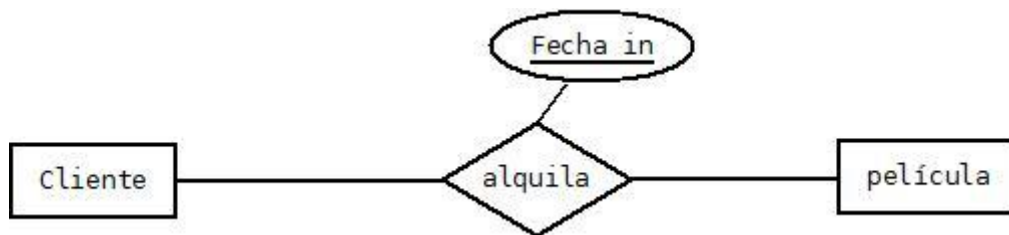
## 2.3 Históricos

Lo utilizamos cuando necesitamos mantener un historial de las ocurrencias de una relación.

**Mal**



**Bien**



### 3.- Transformación a SGBD

Un modelo E/R puede convertirse en un SGBD relacional, para ello debemos seguir una serie de pasos que conviertan nuestro modelo conceptual en diagramas en un modelo relacional, esto se llama paso a tablas. En cada tabla cada intersección fila-columna solo puede tener un elemento.

#### 3.1.- Paso a tablas

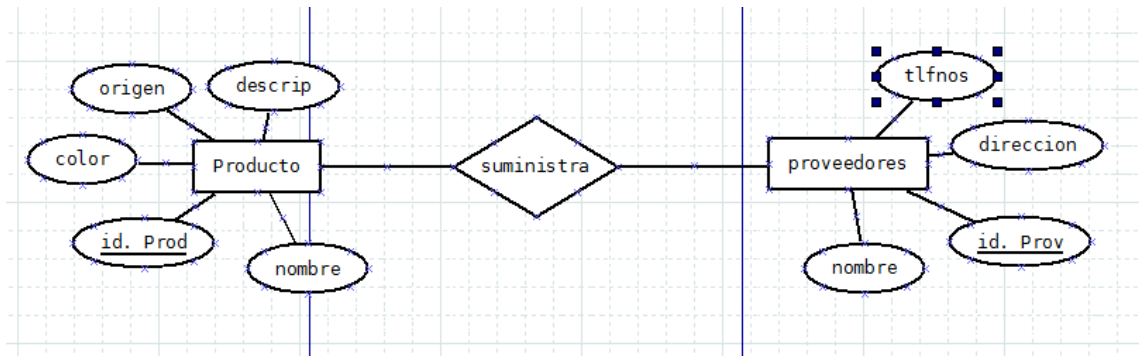
Debemos seguir una serie de reglas:

##### 1. Toda entidad genera una tabla.

Columnas → Atributos de la entidad

PR → Identificador de la entidad

Ejemplo:



Existirán dos tablas:

- Producto (id. Prod, nombre, color, origen, descripción)
- Proveedor (id. Prov, nombre, dirección, tlfno)

El nombre siempre es el de la relación.

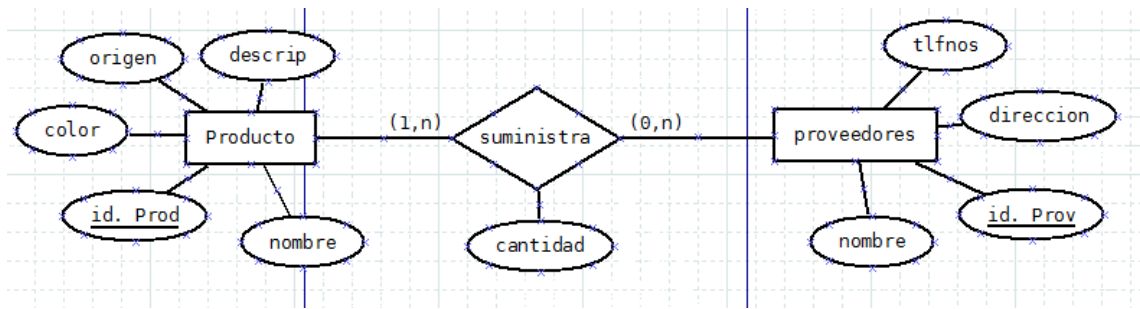
##### 2. Relaciones N:M

Siempre generan tabla

Columnas → Los identificadores de las entidades que participan en la relación. Si tiene atributos forman parte de las columnas.

PR → Identificadores de las entidades participantes en la relación.

Ejemplo:



La tabla generada se denomina igual que la relación. En este caso:

- Suministra (id. Prod, id. Prov, cantidad)

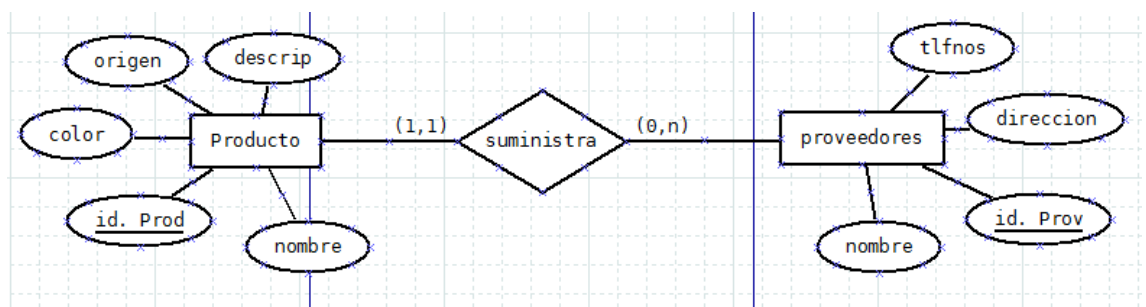
En el paso a tablas se pierden los identificadores.

### 3. Relaciones 1:N

#### 3.1. Sin atributos

No genera tablas. Migra el identificador de la entidad que participa con n a la entidad que participa con 1 como columna.

Ejemplo:



En este caso la relación 'suministra' no genera tabla, tendríamos entonces

- Producto (id. Prod, nombre, color, origen, descripción)
- Proveedor (id. Prov, nombre, dirección, tlfno, id. Prod)

#### 3.2. Con atributos

Siempre genera tabla al igual que una relación N:M

### 4. Relaciones 1:1

#### 4.1. Sin atributos

No genera tabla. Se migra uno de los identificadores de las entidades participantes a la otra.

#### 4.2. Con atributos

Siempre genera tabla como relación N:M

## 5. Atributos multivaluados

Generan una nueva tabla:

Columnas → Identificador de la entidad a la que pertenece. Valor del atributo.

PR → Las columnas que forman la tabla.

## 6. Relaciones grado >2

Siempre generan tabla

Columnas → identificadores de las entidades participantes. Si tiene atributos forma parte de las columnas de la tabla.

PR → Identificadores de las entidades participantes.

## 7. Históricos

Como relaciones grado >2 considerando la fecha como una entidad.

## 8. Jerarquías

### *8.1 Generalización/especialización*

Cada entidad genera tabla:

Padre

Columnas → los atributos de entidad padre más el discriminante de la generalización.

PR → identificador de la entidad.

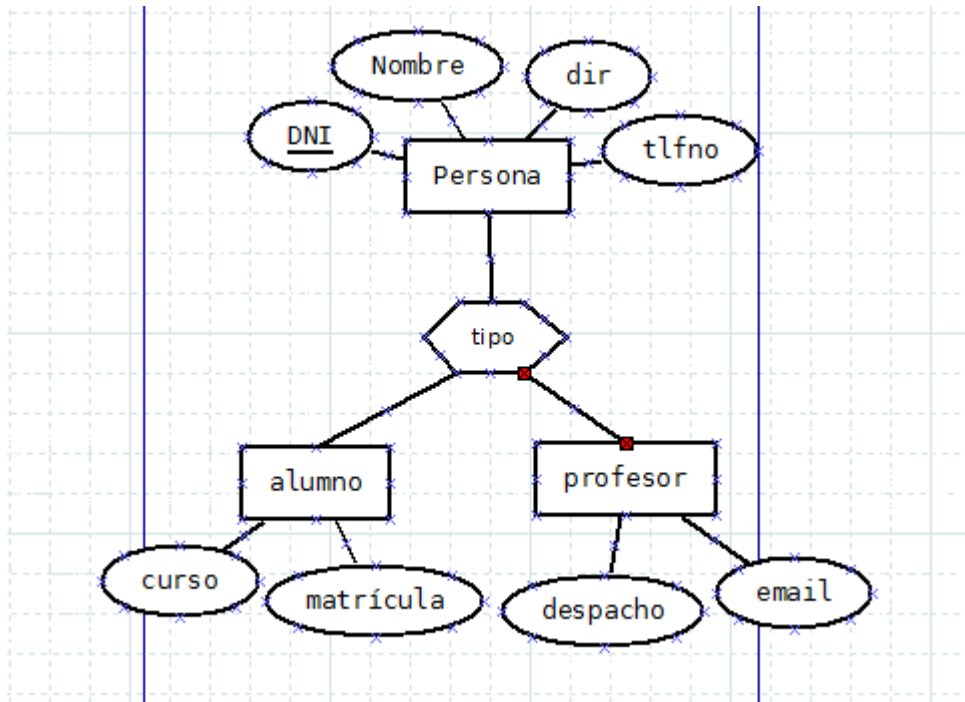
Hijos

Columnas → Los atributos de la entidad hija y el identificador de la entidad padre

PR → identificador de la entidad padre.

Ejemplo:





Tendríamos tres tablas:

- Persona (DNI, nombre, dir, tlfno, tipo)
- Alumno (curso, matrícula, DNI)
- Profesor (despacho, email, DNI)

## 8.2 Subconjunto

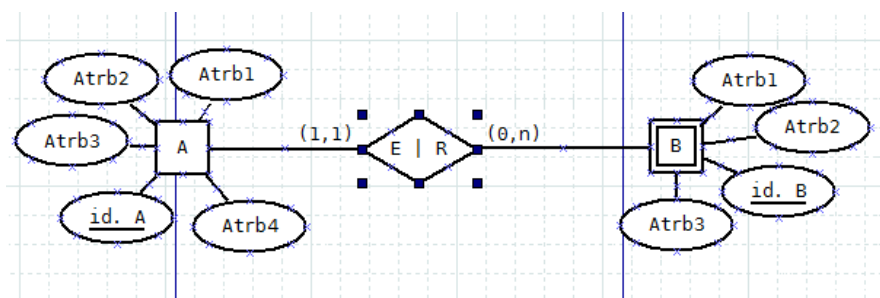
Igual que generalización pero con la diferencia de que la tabla del padre no tendrá discriminante.

## 9. Relaciones de dependencia

### 9.1 Identificador/Existencia(1:N)

No genera tabla. Se tratan como relaciones 1:N. La entidad débil utiliza el identificador de la entidad regular para formar su PR.

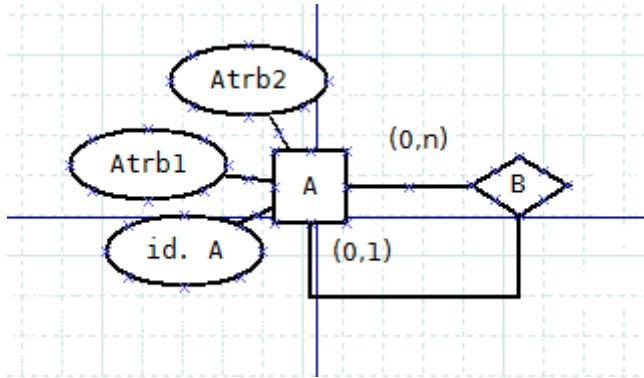
Ejemplo:



Tendríamos dos tablas:

- A (id. A, atrb1, atrb2, atrb3, atrb4)
- B (id. A, atrb1, atrb2, atrb3, id. B)

Ya solo nos queda un caso que no se ha comentado:

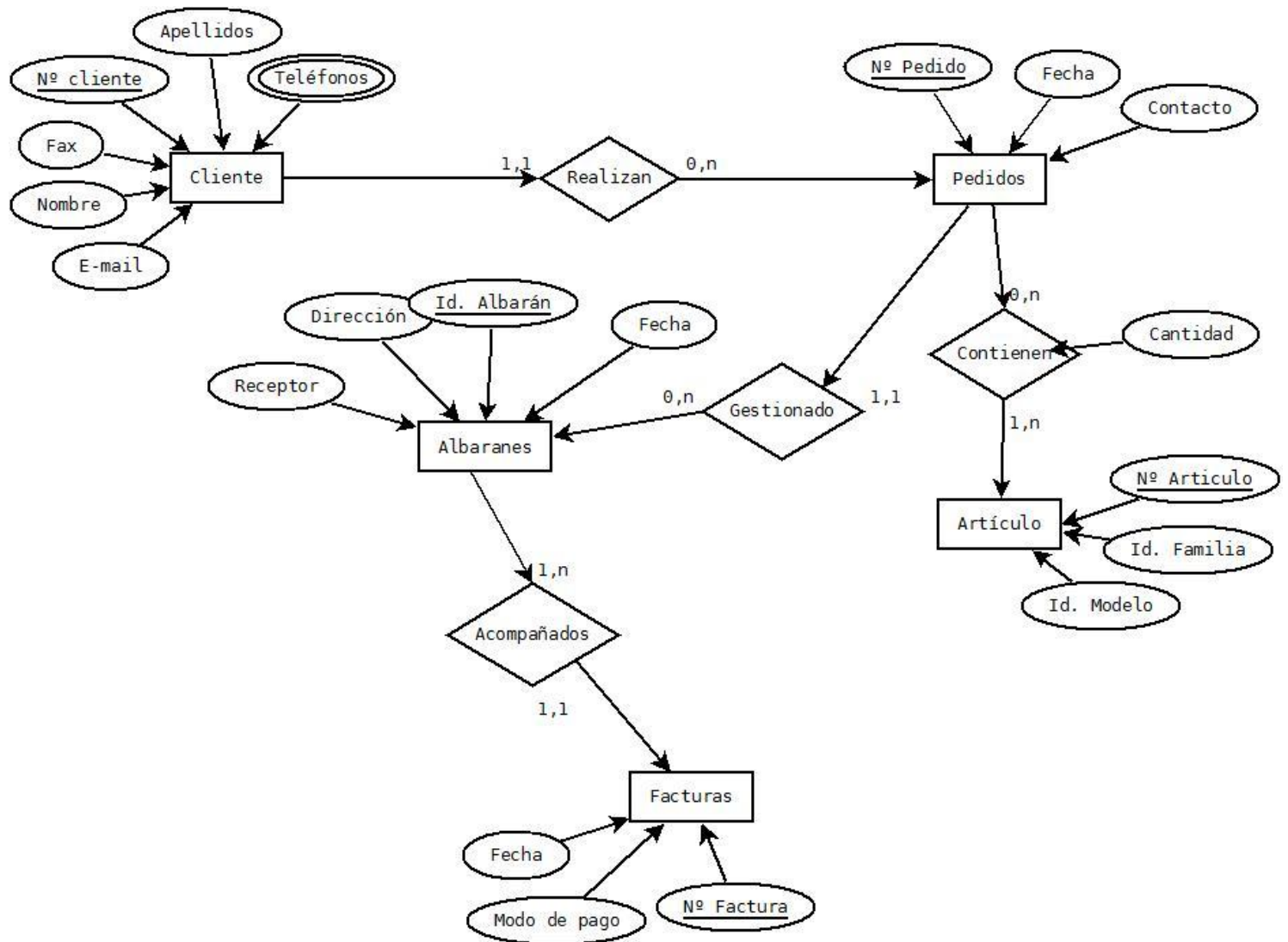


En este caso se crea la tabla:

- A(id. A, atrb1, atrb2, id. A)

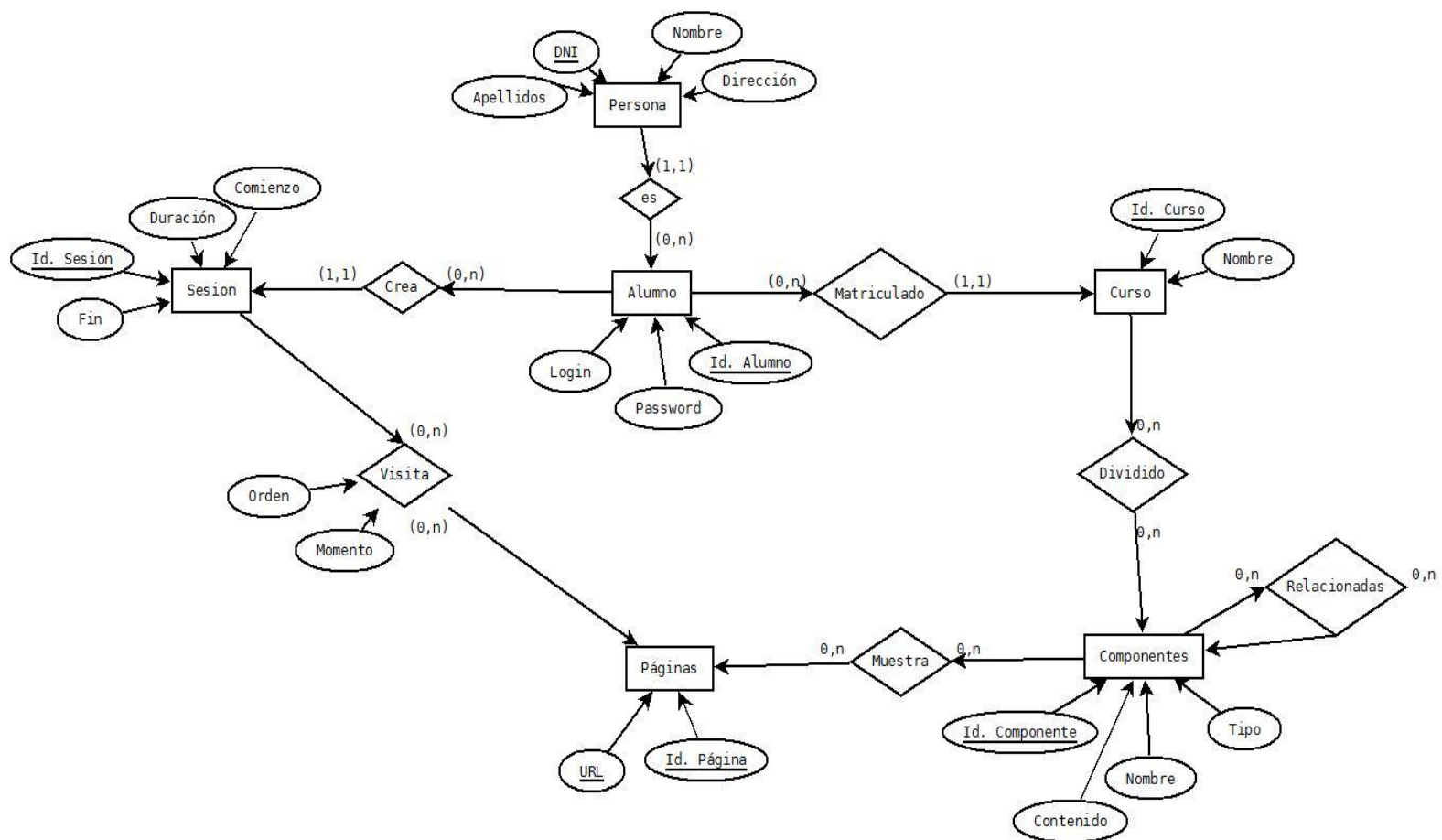
Pero la relación no crea atributo.

## Apéndice A: ejercicio2.dia



Solo debemos tener una consideración con respecto a este ejercicio. Las flechas se deben a una mala comprensión de la herramienta de dibujo y no deben tomarse en cuenta. Debe suponerse las flechas como líneas regulares.

## Apéndice B: ejerciciolunes.dia



Solo debemos tener una consideración con respecto a este ejercicio. Las flechas se deben a una mala comprensión de la herramienta de dibujo y no deben tomarse en cuenta. Debe suponerse las flechas como líneas regulares.